

FILEID**CREATE

N 12

CCCCCCCC CRRRRRRR EEEEEEEE AAAAAA TTTTTTTT EEEEEEEE
CCCCCCCC CRRRRRRR EEEEEEEE AAAAAA TTTTTTTT EEEEEEEE
CC RR RR EE AA AA TT EE
CC RRRRRRRR EEEEEEEE AA AA TT EEEEEE
CC RRRRRRRR EEEEEEEE AA AA TT EEEEEE
CC RR RR EE AAAAAAAA TT EE
CC RR RR EE AAAAAAAA TT EE
CC RR RR EE AA AA TT EE
CC RR RR EE AA AA TT EE
CCCCCCCC RR RR EEEEEEEE AA AA TT EEEEEE
CCCCCCCC RR RR EEEEEEEE AA AA TT EEEEEE

....
....

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLL IIIII SSSSSSS
LLLLLLLLL IIIII SSSSSSS

**F

```
1 0001 0 MODULE CREATE (
2 0002 0           LANGUAGE (BLI<S32),
3 0003 0           IDENT = 'V04-000'
4 0004 0           ) =
5 0005 1 BEGIN
6
7 0007 1
8 0008 1 *****+
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****+
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This module processes the create function. It creates a file with the
38 0038 1 attributes requested, enters it in a directory if desired, and
39 0039 1 accesses it if requested.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1 STARLET operating system, including privileged system services
44 0044 1 and internal exec routines.
45 0045 1
46 0046 1 --
47 0047 1
48 0048 1
49 0049 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 28-Mar-1977 15:05
50 0050 1
51 0051 1 MODIFIED BY:
52 0052 1
53 0053 1 V03-005 LMP0241 L. Mark Pilant, 26-Apr-1984 10:42
54 0054 1           Include the FIB in the MAKE_NAMEBLOCK routine call.
55 0055 1
56 0056 1 V03-004 LMP0154 L. Mark Pilant, 13-Sep-1983 13:17
57 0057 1           Set up a default protection in the created file header
```

: 58 0058 1 from the process default protection.
59 0059 1
60 0060 1 V03-003 ACG0329 Andrew C. Goldstein, 12-Apr-1983 16:05
61 0061 1 Fold long UIC's into [377,377]
62 0062 1
63 0063 1 V03-002 ACG53759 Andrew C. Goldstein, 18-Feb-1983 16:43
64 0064 1 Update revision date and count on ENTER operation
65 0065 1
66 0066 1 V03-001 LMP0018 L. Mark Pilant, 31-Mar-1982 13:10
67 0067 1 Modify to use a local copy of the window complete flag.
68 0068 1
69 0069 1 V02-004 LMP0005 L. Mark Pilant, 29-Dec-1981 14:30
70 0070 1 Added byte limit quota check on window creation, also a
71 0071 1 remap is done if the create did an initial allocation and
72 0072 1 Cathedral windows are desired.
73 0073 1
74 0074 1 V02-003 ACG0247 Andrew C. Goldstein, 23-Dec-1981 1:11
75 0075 1 Set revision date to creation date
76 0076 1
77 0077 1 V02-002 ACG0167 Andrew C. Goldstein, 7-May-1980 18:48
78 0078 1 Previous revision history moved to F11A.REV
79 0079 1 **
80 0080 1
81 0081 1
82 0082 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
83 0083 1 REQUIRE 'SRC\$:FCPDEF.B32';

85 0398 1 GLOBAL ROUTINE CREATE =
86 0399 1 !++
87 0400 1 :
88 0401 1 : FUNCTIONAL DESCRIPTION:
89 0402 1 :
90 0403 1 :
91 0404 1 : This routine processes the CREATE function. It creates a file with the
92 0405 1 : attributes requested, enters it in a directory if desired, and
93 0406 1 : accesses the file if requested.
94 0407 1 :
95 0408 1 : CALLING SEQUENCE:
96 0409 1 : CREATE ()
97 0410 1 :
98 0411 1 : INPUT PARAMETERS:
99 0412 1 : NONE
100 0413 1 :
101 0414 1 : IMPLICIT INPUTS:
102 0415 1 : CURRENT_VCB: VCB of volume
103 0416 1 : IO_PACKET: packet of this I/O request
104 0417 1 :
105 0418 1 : OUTPUT PARAMETERS:
106 0419 1 : NONE
107 0420 1 :
108 0421 1 : IMPLICIT OUTPUTS:
109 0422 1 : PRIMARY_FCB: FCB of file if accessed
110 0423 1 : CURRENT_WINDOW: window of file if accessed
111 0424 1 : USER_STATUS: I/O status block of user
112 0425 1 :
113 0426 1 : ROUTINE VALUE:
114 0427 1 : 1 if successful
115 0428 1 : 0 if error
116 0429 1 :
117 0430 1 : SIDE EFFECTS:
118 0431 1 : File created, blocks allocated, directory modified, file accessed, etc.
119 0432 1 :
120 0433 1 :--
121 0434 1 :
122 0435 2 BEGIN
123 0436 2 :
124 0437 2 LOCAL
125 0438 2 PACKET : REF BBLOCK, ! address of I/O packet
126 0439 2 ABD : REF BBLOCKVECTOR [,ABD\$C_LENGTH],
127 0440 2 : buffer descriptors
128 0441 2 FIB : REF BBLOCK, ! file identification block
129 0442 2 RESULT_LENGTH, : length of result string from ENTER
130 0443 2 RESULT : VECTOR [20, BYTE], ! result string from ENTER
131 0444 2 NAMEBLOCK : BBLOCK [NMB\$C_LENGTH], ! name block to build RAD-50 name
132 0445 2 IDENT_AREA : REF BBLOCK, ! pointer to file header ident area
133 0446 2 PCB : REF BBLOCK, ! Process Control Block address
134 0447 2 ARB : REF BBLOCK, ! access rights block of caller
135 0448 2 MAP_AREA : REF BBLOCK, ! file header map area
136 0449 2 IDX_FCB : REF BBLOCK, ! FCB of index file
137 0450 2 FCB : REF BBLOCK, ! FCB address
138 0451 2 HEADER : REF BBLOCK, ! address of file header
139 0452 2 FUNCTION : BLOCK [1]; ! function code qualifiers
140 0453 2 :
141 0454 2 EXTERNAL

142 0455 2 USER STATUS : VECTOR,
143 0456 2 CURRENT_VCB : REF BBLOCK,
144 0457 2 PRIMARY_FCB : REF BBLOCK,
145 0458 2 CURRENT_WINDOW : REF BBLOCK,
146 0459 2 IO PACKET : REF BBLOCK,
147 0460 2 FILE HEADER : REF BBLOCK,
148 0461 2 NEW_FID,
149 0462 2 HEADER_LBN,
150 0463 2 SUPER_FID : BBLOCK,
151 0464 2 SECOND_FIB : BBLOCK,
152 0465 2 CLEANUP_FLAGS : BITVECTOR,
153 0466 2 SCH\$GL_PCBVEC : REF VECTOR ADDRESSING_MODE (ABSOLUTE); ! PCB vector
154 0467 2
155 0468 2 EXTERNAL ROUTINE
156 0469 2 GET_FIB, get FIB for operation
157 0470 2 CHECK_PROTECT, check file protection
158 0471 2 CREATE_HEADER, create a file ID and header
159 0472 2 CHECKSUM, compute header checksum
160 0473 2 MARK_DIRTY, mark buffer for write-back
161 0474 2 WRITE_HEADER, write file header
162 0475 2 READ_HEADER, read file header
163 0476 2 ENTER, enter file in directory
164 0477 2 COPY_NAME, copy file name to result string
165 0478 2 MAKE_NAMEBLOCK, convert name string into RAD-50 name block
166 0479 2 GET_TIME, get system date and time string
167 0480 2 CREATE_FCB, create an FCB
168 0481 2 CREATE_WINDOW, create a window
169 0482 2 MAKE_ACCESS, complete the access
170 0483 2 MARKDEL_FCB, mark FCB for delete
171 0484 2 WRITE_ATTRIB, write attributes
172 0485 2 EXTEND, extend the file
173 0486 2 SAVE_CONTEXT, save reentrant context area
174 0487 2 RESTORE_CONTEXT, restore reentrant context area
175 0488 2 MARK_DELETE, mark file for delete
176 0489 2 NEXT_HEADER, read next extension file header
177 0490 2 UPDATE_FCB, update contents of FCB
178 0491 2 REMAP_FILE, remap the file completely
179 0492 2
180 0493 2
181 0494 2 ! Enable the deaccess cleanup if an access is taking place.
182 0495 2 !
183 0496 2
184 0497 2 PACKET = .IO_PACKET;
185 0498 2 FUNCTION = .PACKET[IRPSW_FUNC];
186 0499 2 IF .FUNCTION[IOSV_ACCESS]
187 0500 2 THEN
188 0501 3 BEGIN
189 0502 3 CLEANUP_FLAGS[CLF_ZCHANNEL] = 1;
190 0503 3 CLEANUP_FLAGS[CLF_DELWINDOW] = 1;
191 0504 2 END;
192 0505 2
193 0506 2 ! Set up pointers to interesting control blocks.
194 0507 2 !
195 0508 2
196 0509 2 PCB = .SCH\$GL_PCBVEC[.(IO_PACKET[IRPSL_PID])<0,16>];
197 0510 2 ABD = .BBLOCK[.PACKET[IRPSL_SVAPTE], XIBSL_DESCRIPTOR];
198 0511 2 ! pointer to buffer descriptors

```
199 0512 2 FIB = GET_FIB (.ABD); . pointer to FIB
200 0513 2
201 0514 2 IF .FIB[FIB$V_TRUNC]
202 0515 3 OR (.FUNCTION[IOSV_DELETE] AND NOT .FUNCTION[IOSV_ACCESS])
203 0516 3 OR (NOT .FUNCTION[IOSV_CREATE]
204 0517 4 AND (.FIB[FIB$V_EXTEND]
205 0518 4 OR .PACKET[IRPSW_BCNT] GTR ABD$C_ATTRIB
206 0519 4 OR .FUNCTION[IOSV_ACCESS]
207 0520 4 )
208 0521 3 )
209 0522 2 THEN ERR_EXIT (SSS_BADPARAM);
210 0523 2
211 0524 2 IF .CURRENT_VCB[VCB$V_NOALLOC]
212 0525 2 THEN ERR_EXIT (SSS_WRITLCK);
213 0526 2
214 0527 2 ! Do the create if requested. Start by allocating a file number from the
215 0528 2 index file bitmap and reading in the initial file header.
216 0529 2 !
217 0530 2
218 0531 2 IF .FUNCTION[IOSV_CREATE]
219 0532 2 THEN
220 0533 3 BEGIN
221 0534 3 CHECK_PROTECT (CREATE_ACCESS, 0, 0);
222 0535 3
223 0536 3 HEADER = CREATE_HEADER ();
224 0537 3 FIB[FIB$W_FID_NUM] = .HEADER[FH1$W_FID_NUM];
225 0538 3 FIB[FIB$W_FID_SEQ] = .HEADER[FH1$W_FID_SEQ];
226 0539 3 FIB[FIB$W_FID_RVN] = 0;
227 0540 3
228 0541 3 ! Now build an initialized file header in the buffer.
229 0542 3 !
230 0543 3
231 0544 3 HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH / 2;
232 0545 3 HEADER[FH1$B_MP_OFFSET] = (FH1$C_LENGTH+FH1$C_LENGTH) / 2;
233 0546 3 HEADER[FH1$W_STRUCTLEV] = FH1$C_ELEVEL;
234 0547 3 ARB = PACKET[IRPSL_ARB];
235 0548 3 HEADER[FH1$B_UICMEMBER] = .(ARB[ARB$UIC])<0,8>;
236 0549 3 HEADER[FH1$B_UICGROUP] = .(ARB[ARB$UIC])<16,8>;
237 0550 3 IF .(ARB[ARB$UIC])<8,8> NEQ 0
238 0551 3 OR .(ARB[ARB$UIC])<24,8> NEQ 0
239 0552 3 THEN
240 0553 4 BEGIN
241 0554 4 HEADER[FH1$B_UICMEMBER] = -1;
242 0555 4 HEADER[FH1$B_UICGROUP] = -1;
243 0556 3 END;
244 0557 3 HEADER[FH1$W_FILEPROT] = .PCB[PCBSL_DEFPROT];
245 0558 3
246 0559 3 CH$FILL (0, 512 - $BYTEOFFSET(FH1$W_FILECHAR), HEADER[FH1$W_FILECHAR]);
247 0560 3
248 0561 3 IF .FUNCTION[IOSV_DELETE]
249 0562 3 THEN HEADER[FH1$V_MARKDEL] = 1;
250 0563 3
251 0564 3 IF .CLEANUP_FLAGS[CLF_SPOOLFILE]
252 0565 3 THEN HEADER[FH1$V_SPOOL] = 1;
253 0566 3
254 0567 3 MAP_AREA = .HEADER + FH1$C_LENGTH + FH1$C_LENGTH;
255 0568 3 MAP_AREA[FM1$B_COUNTSIZE] = 1;
```

```

256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625

```

```
313 0626 3 THEN
314 0627 4 BEGIN
315 0628 4   FCB = KERNEL_CALL (CREATE_FCB, .HEADER);
316 0629 4   PRIMARY_FCB = .FCB;
317 0630 4
318 P 0631 4   CURRENT_WINDOW = KERNEL_CALL (CREATE_WINDOW, .FIB[FIBSL_ACCTL],
319 0632 4     .FIB[FIB$B_WSIZE], .HEADER, .PACKRET[IRP$L PID], .FCB);
320 0633 4   IF .CURRENT_WINDOW EQ 0 THEN ERR_EXIT (SS$_EXBYTLM);
321 0634 4   KERNEL_CALL (MAKE_ACCESS, .FCB, .CURRENT_WINDOW, .ABD);
322 0635 4   IF .FUNCTION[IOSV-DELETE]
323 0636 4     THEN KERNEL_CALL (MARKDEL_FCB, .FCB);
324 0637 3   END;
325 0638 3
326 0639 3   ! Now extend the file if requested.
327 0640 3
328 0641 3
329 0642 3   IF .FIB[FIB$V_EXTEND] THEN EXTEND (.FIB, .HEADER);
330 0643 3   HEADER = .FILE_HEADER;
331 0644 3   KERNEL_CALL (UPDATE_FCB, .HEADER);
332 0645 2   END;
333 0646 2
334 0647 2 CHECKSUM (.HEADER);
335 0648 2 WRITE_HEADER ();
336 0649 2
337 0650 2 ! Perform the remap operation if necessary to account for any initial extend.
338 0651 2 !
339 0652 2
340 0653 2 ! IF .FUNCTION[IOSV ACCESS] AND .FIB[FIB$V_EXTEND]
341 0654 2 THEN IF .CURRENT_WINDOW[WCB$V_CATHEDRAL]
342 0655 2 THEN REMAP_FILE ();
343 0656 2
344 0657 2 ! If this is a supersede operation, delete the file that was removed during
345 0658 2 the enter operation above. This must be done last since we cannot undo
346 0659 2 a delete in cleaning up from a subsequent error. We first copy the primary
347 0660 2 context into the context save area since this is a secondary operation.
348 0661 2 !
349 0662 2
350 0663 2 IF TESTBITSC (CLEANUP_FLAGS[CLF_SUPERSEDE])
351 0664 2 THEN
352 0665 3 BEGIN
353 0666 3   USER_STATUS[0] = SS$_SUPERSEDE;
354 0667 3   SAVE_CONTEXT ();
355 0668 3   CHSCOPY (FIDSC_LENGTH, SUPER_FID, 0,
356 0669 3     FIBSC_LENGTH - $BYTEOFFSET(FIB$W_FID), SECOND_FIB[FIB$W_FID]);
357 0670 3   MARK_DELETE (SECOND_FIB);
358 0671 3   RESTORE_CONTEXT ();
359 0672 2   END;
360 0673 2
361 0674 2
362 0675 2 RETURN 1;
363 0676 2
364 0677 1 END;
```

! end of routine CREATE

```
.TITLE CREATE
.IDENT \V04-000\
```

				.EXTRN	USER STATUS, CURRENT_VCB
				.EXTRN	PRIMARY_FCB, CURRENT_WINDOW
				.EXTRN	IO_PACKET, FILE_HEADER
				.EXTRN	NEW_FID, HEADER_LBN
				.EXTRN	SUPER_FID, SECOND_FIB
				.EXTRN	CLEANUP_FLAGS, SCRSGL_PCBVEC
				.EXTRN	GET_FIB, CHECK_PROTECT
				.EXTRN	CREATE_HEADER, CHECKSUM
				.EXTRN	MARK_DIRTY, WRITE_HEADER
				.EXTRN	READ_HEADER, ENTER
				.EXTRN	COPY_NAME, MAKE_NAMEBLOCK
				.EXTRN	GET_TIME, CREATE_FCB
				.EXTRN	CREATE_WINDOW, MAKE_ACCESS
				.EXTRN	MARKDEC_FCB, WRITE_ATTRIB
				.EXTRN	EXTEND, SAVE_CONTEXT
				.EXTRN	RESTORE_CONTEXT
				.EXTRN	MARK_DELETE, NEXT_HEADER
				.EXTRN	UPDATE_FCB, REMAP_FILE
				.EXTRN	SYSSCMRRNL
				.PSECT	SCODES,NOWRT,2
			OFFC 00000	.ENTRY	CREATE, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
					R11
		5E	C0	MOVAB	-64(SP), SP
		50	0000G	MOVL	IO_PACKET, R0
		59		MOVL	R0_PACKET
		58	20	MOVZWL	32(PACKET), FUNCTION
		58		BBC	#6, FUNCTION, 1\$
07		CF	0402	BISW2	#1026, CLEANUP_FLAGS+2
		51	00000000G	MOVL	@#SCH\$GL_PCBVEC, R1
		50	OC	MOVZWL	12(R0) R0
		52	6140	MOVL	(R1)[R0], PCB
		5A	2C	MOVL	@44(PACKET), ABD
				PUSHL	ABD
		0000G	CF	CALLS	#1, GET_FIB
		58		MOVL	R0, FIB
		18	17	BLBS	23(FIB), 3\$
		58		BBC	#8, FUNCTION, 2\$
04		13	58	BBC	#6, FUNCTION, 3\$
			5B	TSTB	FUNCTION
			95	BLSS	4\$
			00046	TS1B	22(FIB)
			12	BLSS	3\$
			19	0004A	
			16	TS1B	22(FIB)
			A8	BLSS	3\$
		05	32	(MPW	50(PACKET), #5
			0A	BGTRU	3\$
			19	0004D	
			04	BBC	#6, FUNCTION, 4\$
03		5B	1A	CHMU	#20
			00053	RET	
			06	MOVL	CURRENT_VCB, R0
			E1	BBC	#4, 11(R0), 5\$
		14	00055	CHMU	#604
			04	RET	
05		50	0000G	TSTB	FUNCTION
		A0	CF	BLSS	6\$
		0B	DO	BRW	11\$
			04	CLRQ	-(SP)
		025C	E1		0534
			BF		
			00066		
			04		
			0006A		
			5B		
			95		
			0006B		
			03		
			19		
		008C	0006D		
			31		
			0006F		
		7E	7C		
			00072		
			68:		

0000G	CF	03	DD	00074	PUSHL	#3	
0000G	CF	03	FB	00076	CALLS	#3, CHECK_PROTECT	
	57	00	FB	0007B	CALLS	#0, CREATE_HEADER	
04	A8	02	D0	00080	MOVL	R0, HEADER	0536
		08	A7	00083	MOVL	2(HEADER), 4(FIB)	0537
	67	2E17	8F	00088	CLRW	8(FIB)	0539
06	A7	0101	B0	00090	MOVW	#11799, (HEADER)	0544
	50	58	A9	00096	MOVL	#257, 6(HEADER)	0546
08	A7	38	A0	0009A	MOVW	88(PACKET), ARB	0547
09	A7	3A	A0	0009F	MOVW	56(ARB), 8(HEADER)	0548
		39	A0	000A4	MOVW	58(ARB), 9(HEADER)	0549
			95	000A7	TSTB	57(ARB)	0550
			05	12	BNEQ	7\$	0551
			38	A0	TSTB	59(ARB)	0551
08	A7	FFFC	8F	B0	BEQL	8\$	
0A	A7	0114	C2	B0	MOVW	#65535, 8(HEADER)	0554
00	6E	00	2C	000BA	MOVW	276(PCB), 10(HEADER)	0557
		OC	A7	000C1	MOVCS	#0, (SP), #0, #500, 12(HEADER)	0559
05	58	08	E1	000C3	BBC	#8, FUNCTION, 9\$	
0D	A7	80	8F	000C7	BISB2	#128, 13(HEADER)	0561
		0000G	CF	95	TSTB	CLEANUP_FLAGS	0562
			95	000CC	BGEQ	10\$	0564
0D	A7	04	18	000D0	BISB2	#16, 13(HEADER)	
	50	A7	10	000D2	MOVAB	92(R7), MAP_AREA	0565
06	A0	CC000301	A7	000D6	MOVW	#-872414463, 6(MAP_AREA)	0567
		0000G	8F	000DA	CLRL	NEW_FID	0568
			CF	D4	PUSHL	HEADER	0573
			57	000E2	CALLS	#1, MARK_DIRTY	0574
0000G	CF	01	FB	000E8	BISB2	#32, CLEANUP_FLAGS+2	0575
0000G	CF	20	88	000ED	MOVL	HEADER, FILE_HEADER	0576
0000G	CF	57	D0	000F2	PUSHL	HEADER	0577
0000G	CF	57	DD	000F7	CALLS	#1, CHECKSUM	
		01	FB	000F9	TSTW	10(FIB)	0587
		0A	A8	B5	BEQL	12\$	
			17	13	TSTB	CLEANUP_FLAGS	
			95	00103	BLSS	12\$	
			11	19	PUSHAB	RESULT	
			2C	AE	PUSHAB	RESULT_LENGTH	0589
			04	AE	PUSHL	FIB	
			58	DD	PUSHL	ABD	
			5A	DD	CALLS	#4, ENTER	
			04	FB	BRB	13\$	
			23	11	PUSHL	ABD	
			5A	DD	PUSHL	#1	
			01	DD	PUSHL	SP	
			5E	DD	PUSHAB	COPY_NAME	
00000000G	9F	0000G	CF	9F	CALLS	#4, 3#SYSSCMKRN	
	6E	04	FB	00124	MOVZWL	18(ABD), RESULT_LENGTH	0593
	51	12	AA	3C	MOVAB	16(ABD), R1	0595
	50	10	AA	0012B	MOVZWL	(R1), R0	
			9E	0012F	MOVC3	RESULT_LENGTH, 1(R1)[R0], RESULT	0594
2C	AE	01 A140	61	3C	CLRL	- (SP)	0602
			61	00133	PUSHAB	4(FIB)	
			6E	28	CALLS	#2, READ_HEADER	
			7E	D4	MOVL	RO, HEADER	
			04	A8	PUSHAB	NAMEBLOCK	0603
			02	FB			
			50	D0			
			04	AE			
			9F	00142			
			50	00147			
			04	AE			
			9F	0014A			

			30	AE 9F 0014D	PUSHAB	RESULT		
			08	AE DD 00150	PUSHL	RESULT_LENGTH		
				58 DD 00153	PUSHL	FIB		
				04 FB 00155	CALLS	#4, MAKE_NAMEBLOCK		
				67 9A 0015A	MOVZBL	(HEADER) R0	0604	
66	000CG	CF	6740	3E 0015D	MOVAW	(HEADER)[R0], IDENT_AREA		
	50			0A 28 00161	MOV C3	#10, NAMEBLOCK+6, (IDENT_AREA)	0605	
	0A	AE	0C	A6 9F 00166	PUSHAB	12(IDENT_AREA)	0606	
	56			01 FB 00169	CALLS	#1, GET_TIME		
	0000G	CF		0A A6 B6 0016E	INCW	10(IDENT_AREA)	0607	
				5B 95 00171	TSTB	FUNCTION	0613	
				03 19 00173	BLSS	14\$		
				00AF 31 00175	BRW	19\$		
19	A6	0C	0A	A6 B4 00178	CLR W	10(IDENT AREA)	0616	
	05		32	0D 28 0017B	MOV C3	#13, 12(IDENT AREA), 25(IDENT_AREA)	0617	
				A9 B1 00181	CMPW	50(PACKET), #5	0619	
				0B 1B 00185	BLEQU	15\$		
				7E D4 00187	CLRL	-(SP)	0620	
6D	0000G	CF	0480	8F BB 00189	PUSHR	#^M<R7, R10>		
	5B			03 FB 0018D	CALLS	#3, WRITE_ATTRIB		
				06 E1 00192	BBC	#6, FUNCTION, 17\$	0625	
				57 DD 00196	PUSHL	HEADER	0628	
				01 DD 00198	PUSHL	#1		
				5E DD 0019A	PUSHL	SP		
	00000000G	9F	0000G	CF 9F 0019C	PUSHAB	CREATE_FCB		
	52			04 FB 001A0	CALLS	#4, @#SYSSCMKRNL		
	0000G	CF		50 DD 001A7	MOVL	R0, FCB		
				52 DD 001AA	MOVL	FCB, PRIMARY_FCB	0629	
				52 DD 001AF	PUSHL	FCB	0632	
			0C	A9 DD 001B1	PUSHL	12(PACKET)		
				57 DD 001B4	PUSHL	HEADER		
			7E	03 A8 98 001B6	CVTBL	3(FIB), -(SP)		
				68 DD 001BA	PUSHL	(FIB)		
				05 DD 001BC	PUSHL	#5		
				5E DD 001BE	PUSHL	SP		
	00000000G	9F	0000G	CF 9F 001C0	PUSHAB	CREATE_WINDOW		
	0000G	CF		08 FB 001C4	CALLS	#8, @#SYSSCMKRNL		
				50 DD 001CB	MOVL	R0, CURRENT_WINDOW		
				05 12 001D0	BNEQ	16\$	0633	
			2A14	8F BF 001D2	CHMU	#10772		
				04 001D6	RET			
				5A DD 001D7	16\$:	PUSHL	ABD	0634
				CF DD 001D9	PUSHL	CURRENT_WINDOW		
				52 DD 001DD	PUSHL	FCB		
				03 DD 001DF	PUSHL	#3		
				5E DD 001E1	PUSHL	SP		
11	00000000G	9F	0000G	CF 9F 001E3	PUSHAB	MAKE_ACCESS		
	5B			06 FB 001E7	CALLS	#6, @#SYSSCMKRNL		
				08 E1 001EE	BBC	#8, FUNCTION, 17\$	0635	
				52 DD 001F2	PUSHL	FCB	0636	
				01 DD 001F4	PUSHL	#1		
				5E DD 001F6	PUSHL	SP		
	00000000G	9F	0000G	CF 9F 001F8	PUSHAB	MARKDEL_FCB		
				04 FB 001FC	CALLS	#4, @#SYSSCMKRNL		
			16	A8 95 00203	TSTB	22(FIB)	0642	
				09 18 00206	BGEQ	18\$		
				57 DD 00208	PUSHL	HEADER		

		0000G	CF	57	0000G	58	DD 0020A	PUSHL	FIB		
						02	FB 0020C	CALLS	#2, EXTEND		
						CF	DD 00211	18\$:	FILE HEADER, HEADER		
						57	DD 00216	PUSHL	HEADER		
						01	DD 00218	PUSHL	#1		
						5E	DD 0021A	PUSHL	SP		
		00000000G	9F		0000G	CF	9F 0021C	PUSHAB	UPDATE FCB		
						04	FB 00220	CALLS	#4, @#SYSSCMKRNL		
						57	DD 00227	19\$:	PUSHL	HEADER	
		14	0000G	CF		01	FB 00229	CALLS	#1, CHECKSUM		
			0000G	CF		00	FB 0022E	CALLS	#0, WRITE HEADER		
				58		06	E1 00233	BBC	#6, FUNCTION, 20\$		
						A8	95 00237	TSTB	22(FIB)		
						OF	18 0023A	BGEQ	20\$		
		05	08	A0	0000G	CF	0023C	MOVL	CURRENT WINDOW, R0		
			0000G	CF		06	E1 00241	BBC	#6, 11(R0), 20\$		
		24	0000G	CF		00	FB 00246	CALLS	#0, REMAP FILE		
			0000G	CF	0631	05	E5 0024B	BBCC	#5, CLEANUP_FLAGS, 21\$		
						8F	3C 00251	MOVZWL	#1585, USER_STATUS		
		3C	00	0000G	CF	00	FB 00258	CALLS	#0, SAVE CONTEXT		
						06	2C 0025D	MOVC5	#6, SUPER_FID, #0, #60, SECOND_FIB+4		
						0000G	CF 00264	PUSHAB	SECOND_FIB		
						0000G	CF 00267	CALLS	#1, MARK_DELETE		
						01	FB 0026B	CALLS	#0, RESTORE_CONTEXT		
						0000G	CF 00270	MOVL	#1, R0		
						50	01 DD 00275	21\$:	RET		
							04 00278				

: Routine Size: 633 bytes, Routine Base: \$CODE\$ + 0000

```
: 365 0678 1
: 366 0679 1 END
: 367 0680 0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	633	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----	Pages Mapped	Processing Time
	Total Loaded Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619 62 0	1000	00:01.9

CREATE
V04-000

M 13
16-Sep-1984 00:53:11
14-Sep-1984 12:29:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]CREATE.B32;1 Page 12
(2)

CR
VO

: COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS\$:CREATE/OBJ=OBJ\$:CREATE MSRC\$:CREATE/UPDATE=(ENH\$:CREATE)

: Size: 633 code + 0 data bytes
: Run Time: 00:16.3
: Elapsed Time: 00:46.9
: Lines/CPU Min: 2506
: Lexemes/CPU-Min: 17561
: Memory Used: 240 pages
: Compilation Complete

0164 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

